

(19)



JAPANESE PATENT OFFICE

PATENT ABSTRACTS OF JAPAN

(11) Publication number: 07028633 A

(43) Date of publication of application: 31.01.95

(51) Int. Cl

G06F 9/06
G06F 12/00
G06F 12/00
G06F 13/00

(21) Application number: 05172846

(22) Date of filing: 13.07.93

(71) Applicant: HITACHI LTD

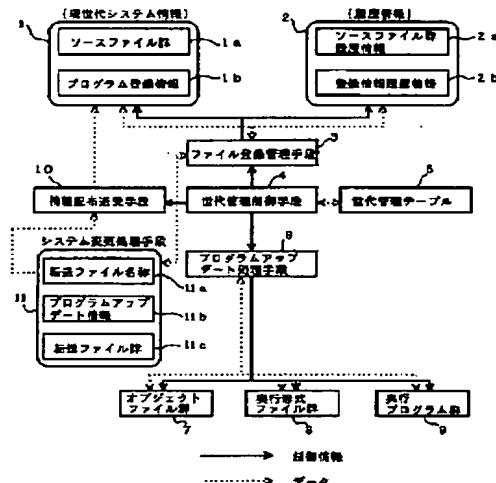
(72) Inventor: NAKAI KOJI
TOBE AKIHIKO
SASAKI RIICHIRO
MIWA NAOKI

(54) METHOD AND DEVICE FOR PROGRAM GROUP
GENERATION MANAGEMENT, PROGRAM
UPDATE PROCESSING METHOD, AND
DECENTRALIZED PROCESSING SYSTEM

(57) Abstract:

PURPOSE: To provide an automated generation management system for the program group of a large-scale system which changes the generation of the program group with high efficiency and high reliability.

CONSTITUTION: A generation management control means 4 refers to a history management table 5 wherein the mutual relation between a system generation and a program generation is defined, and generates a file generation or program registration information generation different from current generation system information 1 by actuating a file registration management means 3 when the change destination system generation is registered in history information 2, thereby substituting the generated generation for the current generation system information 1. When the change destination system generation is an unregistered new generation, it is registered in the current generation system information 1 and history information is generated by the means 3 and registered. For a generation change of only program registration information, a program update means 6 updates only files and programs which need to be registered again.



A

【発行国】日本国特許庁 (JP)

【公報種別】公開特許公報

【公開番号】特開平7-28633 平成7年(1995)1月31日

【発明の名称】プログラム群世代管理方法と装置およびプログラムアップデート処理方法および分散処理システム

【国際特許分類第6版】

(IPC, 識別, 分冊, 庁内No., 技術箇所)

G06F 9/06, 410, Q, 9367-5B,

G06F 12/00, 517, , 8944-5B,

G06F 12/00, 545, A, 8944-5B,

G06F 13/00, 351, H, 7368-5B,

【審査請求】未請求

[F I]

(FI, 識別, 分冊, 庁内No., 技術箇所)

G06F 9/06, 410, Q, 9367-5B,

G06F 12/00, 517, , 8944-5B,

G06F 12/00, 545, A, 8944-5B,

G06F 13/00, 351, H, 7368-5B,

【請求項の数】 13

【出願形態】 O L

【全頁数】 18

【出願番号】特願平5-172846 平成5年(1993)7月13日

【出願人】株式会社日立製作所(000005108)

東京都千代田区神田駿河台四丁目6番地

【発明者】中井 耕治

茨城県日立市大みか町五丁目2番1号 株式会社日立製作所大みか工場内

・戸辺 昭彦

茨城県日立市大みか町五丁目2番1号 株式会社日立製作所大みか工場内

・佐々木 利一郎

茨城県日立市大みか町五丁目2番1号 株式会社日立製作所大みか工場内

・三和 直樹

茨城県日立市大みか町五丁目2番1号 株式会社日立製作所大みか工場内

【代理人】[弁理士]高橋 明夫 (外1名)

【要約】

【目的】

大規模システムにおけるプログラム群を、高効率、高信頼に世代変更できる自動化されたプログラム群の世代管理方式を提供することにある。

【構成】

世代管理制御手段4は、システム世代とプログラム世代の相互関係を定義する履歴管理テーブル5を参照し、変更先システム世代が履歴情報2に登録のある場合は、現世代システム情報2と異なるファイル世代またはプログラム登録情報世代について、ファイル登録管理手段3を起動して生成し、現世代情報1を置換する。

変更先システム世代が未登録の新世代の場合は、現世代システム情報1に登録後、手段3で履歴情報を生成し、登録する。

プログラム登録情報のみの世代変更にたいしては、プログラムアップデート手段6が再登録の必要なファイル/プログラムのみアップデート処理する。

【特許請求の範囲】

【請求項1】

現世代のシステムを構成するプログラム群とプログラム毎の世代の履歴情報を登録し、システムの変更を管理するプログラム群世代管理方法において、前記現世代のシステムまたは前記履歴情報に登録されている複数のプログラムと複数のシステムの相互関係を、プログラム世代のIDによって予め定義し、システムの変更要求にしたがい、前記現世代と変更先のシステムを前記相互関係に基づいてプログラム毎に比較し、両者のプログラム世代IDが一致しないときに変更先のプログラム世代を前記履歴情報から生成し、この生成されたプログラムを現世代の前記プログラム群の中の該当プログラムと置換して、システムの世代変更を行うことを特徴とするプログラム群世代管理方法。

【請求項2】

請求項1において、前記プログラムはソースファイルであり、前記置換によって更新されたソースファイルを元に、それをコンパイルしたオブジェクトファイル、それをリンクロードした実行形式ファイル、それをメモリ上に記憶した実行プログラムを順次再登録する、アップデート処理が行なわれることを特徴とするプログラム群世代管理方法。

【請求項3】

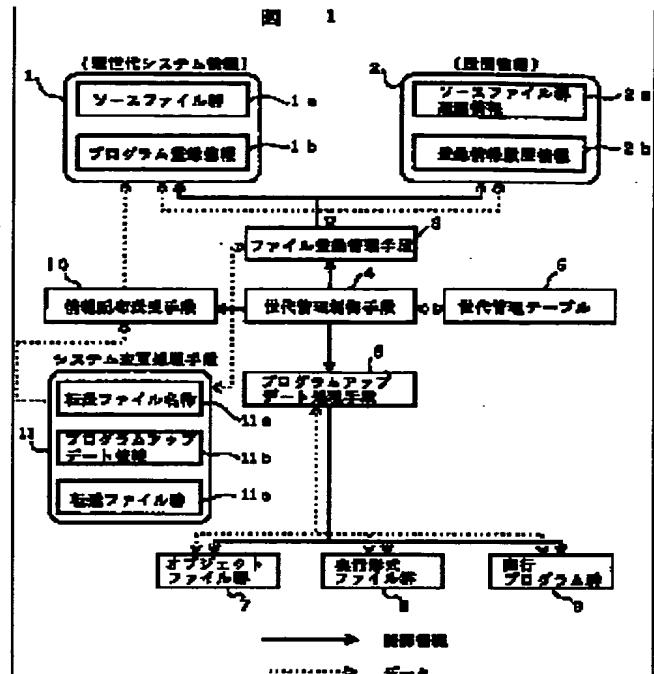
請求項1または2において、前記変更先のシステムが前記履歴情報に登録の無い新世代プログラムを含むとき、前記新世代プログラムに対応する前記履歴情報の最新プログラムを生成し、前記新世代プログラムと前記最新プログラムを照合して新たな履歴情報を生成することを特徴とするプログラム群世代管理方法。

【請求項4】

プログラム毎の世代変更と、複数のプログラムの組合せで構成されるシステムの世代変更を管理するプログラム群世代管理装置において、各プログラム毎にその履歴情報を記憶する履歴情報記憶手段と、現世代のシステムを構成する複数のソースファイルからなるプログラム群を記憶する現世代システム情報記憶手段と、所定の履歴情報から所定のソースファイルを生成し、前記現世代システム情報記憶手段に登録または置換するファイル登録管理手段と、前記履歴情報に登録されているプログラムとシステムの相互関係を、プログラム世代IDによって定義する世代管理テーブルと、変更先のシステムを指示する入力手段からの変更要求にしたがい、現世代と変更先のシステム世代を前記世代管理テーブルを参照してプログラム毎に比較し、両者のプログラム世代IDが一致しないときに前記ファイル履歴手段を動作させて変更先のプログラム世代を前記履歴情報から生成し、現世代システムの該当プログラムと置換させる世代管理制御手段と、を備えることを特徴とするプログラム群世代管理装置。

【請求項5】

請求項4において、前記現世代システム情報記憶手段には前記ソースファイルのプログラム登録情報が登録さ



れ、前記履歴情報記憶手段には各プログラム世代のプログラム登録情報に関する履歴情報が記憶され、前記世代管理テーブルにはシステム世代とプログラム登録情報の相互関係を、プログラム登録情報世代IDによって定義されることを特徴とするプログラム群世代管理装置。

【請求項6】

請求項5において、前記現世代システム情報記憶手段に登録されているプログラム登録情報が変更される場合に、その変更内容に応じてオブジェクトファイルまたは実行形式ファイルまたは実行プログラムのいずれかから再登録を開始するプログラムアップデート処理手段を備えることを特徴とするプログラム群世代管理装置。

【請求項7】

ソースファイルが更新される場合に、オブジェクトファイル、実行形式ファイルおよび実行プログラムの順番でプログラム登録情報に基づく再登録を行うプログラムアップデート処理方法において、ソースファイルに対応するプログラム登録情報のみが変更される場合は、再登録する前記順番のもので、前記変更により影響を受けるもの以後を再登録することを特徴とするプログラムアップデート処理方法。

【請求項8】

請求項7において、前記プログラム登録情報にはコンパイルオプション、リンクコードオプション、実行プログラムオプションの少なくとも一つを含み、コンパイルオプションが変更されるときはオブジェクトファイルから再登録し、リンクコードオプションが変更されるときは実行形式ファイルから再登録し、実行プログラムオプションが変更されるときは実行プログラムを再登録することを特徴とするプログラムアップデート処理方法。

【請求項9】

請求項7または請求項8において、前記プログラム登録情報の変更内容に応じて、前記変更により影響を受けるものの一つ前の順番のものの登録日付を更新することを特徴とするプログラムアップデート方法。

【請求項10】

請求項7または請求項8において、前記プログラム登録情報の変更内容に応じて、前記変更により影響を受けるもの以後の現登録日付を削除することを特徴とするプログラムアップデート方法。

【請求項11】

世代管理機能をもつマスタ計算機と世代管理機能をもたない他の計算機群をネットワークで接続する分散処理システムにおいて、マスタ計算機は、複数のソースファイルとそのプログラム登録情報を記憶する現世代システム情報記憶手段と、前記ソースファイルおよび/またはプログラム登録情報を更新し、かつ、更新したソースファイルまたはプログラム登録情報または削除するファイル名称の少なくとも一つを含むシステム変更情報を生成するプログラム登録管理手段と、前記システム変更情報を他の計算機に配布する送信手段を備え、前記他の計算機は、マスタ計算機からのシステム変更情報を受信する受信手段と、自己の現世代システム情報記憶手段と、受信したソースファイルまたはプログラム登録情報または削除するファイル名称に従って前記自己の現世代システム情報記憶手段を更新するプログラム登録管理手段を備えることを特徴とする分散処理システム。

【請求項12】

請求項11において、前記マスタ計算機のプログラム登録管理手段は、前記プログラム登録情報の変更内容にしたがい、対応するソースファイル毎にオブジェクトファイル、実行形式ファイルおよび実行プログラムの順番で行うプログラムアップデート処理の開始位置を前記システム変更情報に付加し、前記他の計算機は、前記開始位置に示されるファイル/プログラムから変更されたプログラム登録情報に対応するソースファイルのアップデート処理を行うプログラムアップデート処理手段を備えることを特徴とする分散処理システム。

【請求項13】

請求項11または請求項12において、分散処理システム内の同一種類の計算機はそれを管理するマスタ計算機と同時に世代変更されることを特徴とする分散処理システム。

【発明の詳細な説明】

【0001】

【産業上の利用分野】

本発明は、多数のファイル群からなるプログラム群管理システムに係わり、特にシステム世代とファイル世代を統一的に管理する世代管理方式及び、世代変更されたファイルのアップデート方式に関する。

【0002】

【従来の技術】

ソフトウェアの世代(バージョン)管理は、ファイル単位の管理が一般的であり、世代間の差分情報を管理することで履歴情報を削減をしている。

【0003】

多数の計算機を接続したネットワークシステムなどの大規模システムにおいては、多数のファイルが登録されていて、一部のファイルの更新や、利用ファイルの組み合わせ、即ち、ソフトウェアのシステム変更がしばしば行われる。

【0004】

しかし、ファイル単位の履歴情報のみでは、ファイル世代とシステム世代の対応関係が管理されていないために、かかる世代変更はユーザのマニュアル処理に委ねられていた。

【0005】

この対策として、システム世代の変更に伴って全ファイルを同期して更新する方法があるが、変更のないソースファイルについても無条件に再生成、再登録がなされ、さらにアップデート処理が実行される。このため、多数のファイルをもつシステムにおいては、世代変更時のオーバヘッドが大きく、保守の停止時間の短いフル稼働システムには適用し難いものであった。

【0006】

世代変更を効率的に行う方法として、システム世代毎に構成するファイルの版管理を行い、新システムのインストールに際して、旧システムと同じファイルはパスし、異なるファイルはユーザに更新の可否を問い合わせする管理方式が、特開平2-41523号公報に提案されている。

【0007】

一方、世代変更に伴うプログラムのアップデートは、ソースファイルの変更によって起動され、オブジェクト、実行ファイル及び実行プログラムの各段階の登録日付けの違いより、再登録の必要のあるファイル及びプログラムを決定している。

【0008】

【発明が解決しようとする課題】

しかし、上記提案はインストールに限られ、現世代のシステム構成と再構成する変更先世代の同期した管理が行わないので、世代変更におけるユーザの手間は軽減されず、信頼性の上でも問題がある。

【0009】

さらに、従来のアップデート方式では、プログラム登録情報のみの変更によってはアップデートが実行されない。このため、たとえばコンパイル情報の変更がある場合に、対応するオブジェクトファイルの再登録をユーザが改めて指示する必要があった。このため、大規模システムにおいては、ユーザの負担が大きく、指示漏れによる信頼性低下の原因にもなっていた。

【0010】

本発明の第1の目的は、従来技術の欠点を克服し、変更先世代を含むシステム世代とファイル世代を統一的に管理して、世代変更の自動化と処理のオーバヘッドを低減できるプログラム群世代管理方式を提供することにある。

【0011】

本発明の第2の目的は、プログラム登録情報の変更によるプログラムの再登録を自動化するプログラムアップデート方式を提供することにある。

【0012】

本発明の第3の目的は、マスター計算機より配布される一のシステム変更情報によって、システム内の他の計算機のシステムを一斉に変更できる分散処理システムを提供することにある。

【0013】

本発明のその他の目的は、以下の記載を通じて明確にされる。

【0014】

【課題を解決するための手段】

上記第1の目的は、現世代のシステムを構成するプログラム群とプログラム毎の世代（版）の履歴情報を登録し、システムの変更を管理するプログラム群世代管理方法において、前記現世代のシステムまたは前記履歴情報に登録されている複数のプログラムと複数のシステムの相互関係を、プログラム世代のIDによって予め定義し、システムの変更要求にしたがい、前記現世代と変更先のシステムを前記相互関係に基づいてプログラム毎に比較し、両者のプログラム世代IDが一致しないときに変更先のプログラム世代を前記履歴情報から生成し、この生成されたプログラムを現世代の前記プログラム群の中の該当プログラムと置換して、システムの世代変更を行うことにより達成される。

【0015】

上記第2の目的は、ソースファイルが更新される場合に、オブジェクトファイル、実行形式ファイルおよび実行プログラムの順番でプログラム登録情報に基づく再登録を行うプログラムアップデート処理方法において、ソースファイルに対応するプログラム登録情報のみが変更される場合は、再登録する前記順番のもので、前記変更により影響を受けるもの以後を再登録することにより達成される。

【0016】

上記第3の目的は、世代管理機能をもつマスター計算機と世代管理機能をもたない他の計算機群をネットワークで接続する分散処理システムにおいて、マスター計算機は、複数のソースファイルとそのプログラム登録情報を記憶する現世代システム情報記憶手段と、前記ソースファイルおよび／またはプログラム登録情報を更新し、かつ、更新したソースファイルまたはプログラム登録情報または削除するファイル名称の少なくとも一つを含むシステム変更情報を生成するプログラム登録管理手段と、前記システム変更情報を他の計算機に配布する送信手段を備え、前記他の計算機は、マスター計算機からのシステム変更情報を受信する受信手段と、自己の現世代システム情報記憶手段と、受信したソースファイルまたはプログラム登録情報または削除するファイル名称に従って前記自己の現世代システム情報記憶手段を更新するプログラム登録管理手段を備えることにより達成される。

【0017】

【作用】

第1の発明によれば、世代管理テーブルの現世代と変更先世代のファイル世代IDは瞬時に比較でき、一致するときはファイルの生成が省略でき、一致しないとき履歴情報を元に変更先ファイルを生成するので、ファイル生成を含む世代変更が変更先世代IDを指示するのみで自動化でき、かつ、処理のオーバヘッドが大幅に削減できる。

【0018】

第2の発明によれば、ソースファイルは変わらずにプログラム登録情報のみが変更する場合に、その変更内容がたとえば、コンパイル情報であればオブジェクトファイル、リンクコード情報であれば実行形式ファイル、実行プログラム登録情報であれば実行プログラムを、それぞれ再登録の開始位置とするプログラムアップデート処理が実行される。

【0019】

これによって、プログラムアップデートの完全な自動処理が実現でき、従来、ユーザが行っていたプログラム登録情報の変更チェックによる対応ソースファイルの日付更新の手間が省力化でき、チェック漏れによる再登録ミスも防止できる。

さらに、世代変更のための計算機の中断を少なくでき、オンライン業務の処理効率を向上できる。

【0020】

第3の発明によれば、マスター計算機からただ1回のシステム変更情報を送信するのみで、システム内の他の計算機は世代変更のための特別の機能をもつことなく、短時間の中断でオンラインの世代変更が実現でき、分散処理システムの稼働率を向上できる。

【0021】

【実施例】

以下、図面を参照して本発明の実施例を詳細に説明する。

図2は、本発明によるプログラム群の世代変更に好適な分散システムの構成を示す。

複数の計算機101a～101dがネットワークに接続され、各計算機のプログラム群とその変更は、マスター計算機101aによって管理されている。

【0022】

マスター計算機101aは、磁気ディスク(MD)109等に格納されているプログラム群やデータなど登録済のソフトウェアを、入出力手段107を通じて読み出し、現在使用するプログラム群等の組み合わせ(以下、現世代システムと呼ぶ)を世代管理装置104で管理し、この現世代システムの関連情報を通信装置102を介して各計算機に伝送している。

なお、システムの世代変更や新規登録は、ユーザが端末110から指示あるいは入力する。

【0023】

図1は、マスター計算機101aに具備される、世代管理装置104の機能ブロック図で、実線は制御情報、点線はデータの流れを示す。

現世代システム情報記憶手段1は、現世代のアプリケーションソフトを構成するソースファイル群1aと、そのプログ

ラム登録情報 1 b を格納する。

履歴情報記憶手段 2 は、登録済みのソースファイルについて、各ファイルおよびプログラム登録情報毎に、世代（バージョン）間のソースファイル履歴情報 2 a、登録情報履歴情報 2 b を、世代順に格納する。
これら記憶手段 1、2 は、磁気ディスク 109 に記憶領域を有している。

【0024】

ファイル登録管理手段 3 は、現世代記憶手段 1 と履歴情報記憶手段 2 の登録、更新、削除を行うもので、ファイルやプログラム登録情報の世代間の内部データの差分情報をとって、履歴情報 2 a, 2 b を作成する。
また、この履歴情報より所望の現世代ファイル群 1 a およびプログラム登録情報 1 b 生成し、記憶手段 1 への登録と現世代に不要なファイルの削除を行う。

【0025】

プログラムアップデート手段 6 は、新規に登録する場合はソースファイルを元に順次、オブジェクトファイルを記憶手段 7 に、実行形式ファイルを記憶手段 8 に、実行プログラムをメモリ 9（主メモリ）に登録する。
記憶手段 7 と記憶手段 8 は、それぞれ磁気ディスク 109 に記憶領域を有している。

【0026】

一方、更新されたソースファイルを元にアップデートする場合は、各段階のファイル／プログラムの登録日付の新旧比較により連鎖的に実行する。
すなわち、ソースファイル 1 a の登録日付が更新されると、オブジェクトファイル 1 a' の日付の方が古くなるので、ソースファイル 1 a をコンパイルしてオブジェクトファイル 1 a' を再登録する。
すると、ライブラリとして記憶されている実行形式ファイル 1 a'' の日付が古くなるので、変更されたオブジェクトファイル 1 a' をリンクロードして実行形式ファイル 1 a'' を再登録する。
この結果、実行プログラム 1 a の登録日付が古くなり、更新された実行形式ファイル 1 a'' の一部または全部のプログラムが再登録される。

【0027】

世代管理体制御手段 4 は、図 3 に示す世代管理テーブル 5 を参照しながらシステムの世代変更を行う。
世代管理テーブル 5 はシステム単位のシステム世代 ID (SYS 1, SYS 2, . . .) と、履歴情報記憶手段 2 に登録されているファイル単位 (A～F) のファイル世代 ID (V 1, V 2, V 3, . . .) の対応関係を登録するもので、各世代のシステムを構成するファイル群の同期をとるために用いられる。

【0028】

図 3 で、記号「→」は、当該システム世代でファイル世代の変更がないことを示し、前システム世代と同じファイル世代を採用している。
記号「—」は、当該システム世代に当該ファイルを使用していないことを示し、たとえば、ファイル C は履歴情報 2 a には登録されているが、使用は SYS 5 が初めてとなる。

【0029】

同図 (b) の SYS 6 は、同図 (a) では全て空白であり、このシステムは履歴情報記憶手段 2 へ登録のないファイルまたはファイル世代を含む新世代システムである。
なお、「区分」の項目は、システム世代の現状のステータスを示し、CUR は現世代システム、CHG は変更先世代システムを表す識別子である。

【0030】

世代管理テーブル 5 は、システム世代とプログラム登録情報の世代 (V 1, V 2, . . .) との対応関係も定義している。
図 4 (a) ～(d) は、世代管理テーブル 5 の付属情報として定義されるプログラム登録情報テーブル 5 1 で、任意のプログラム、例えばプログラム A のプログラム登録情報について、コンパイルオプション、リンク・ロードオプション、実行プログラムオプション及び全体削除の例を示している。

【0031】

なお、図 3 のソースファイルと図 4 のプログラム情報が対応するプログラム、例えばファイル A とプログラム A は、説明を簡単にするため以下では 1 対 1 に対応しているものとするが、これに限られるものではない。
たとえば、複数のファイル A～C からプログラム A が生成される場合は、プログラム構成情報に基づいて、プログラムアップデート処理が行なわれることは周知であり、本実施例においても適用可能である。

【0032】

次に、システム世代変更に伴う世代管理体制御手段 4 におけるプログラム群の管理を、図 5～図 10 を参照しながら説明する。
なお、ここでのシステム世代変更は、変更先システムを構成するソースファイル等の履歴情報 2 による生成と現世代システム情報 1 への登録であり、さらには、世代変更で更新したソースファイルまたはプログラム登録情報を元にアップデートし、動的な情報である実行形式ファイル及び実行プログラムの再登録である。

【0033】

図 5 は、世代管理の概略を示す処理フローである。

ステップ S 10 で、システム変更要求を、端末 110 からの変更先システム世代 ID により受け付け、ステップ S 20 で、受け付けたシステム世代 ID が世代管理テーブル 5 に登録されているか判定する。

登録されている世代 ID、たとえば SYS 5 への世代変更要求の場合は、世代管理テーブル 5 の SYS 5 に CHG を設定し、後述するステップ S 30 による世代変更処理を行う。

【0034】

一方、ステップ S 10～20 で未登録の世代 ID である SYS 6 が受け付けられた場合は、ステップ S 40 で世代管理テーブル 5 に SYS 6 の ID を追加して区分を CHG とし、未登録のファイルについて端末から 110 からの入力を受け付け、ファイル登録手段 3 を動作させて現世代システム情報記憶手段 1 に登録する。

図 3 (b) の例では、ファイル A の V 5、ファイル D の V 4、ファイル F の V 1 および登録情報の V 3 が新規登録される。

この後、新世代登録処理 S 50 で、後述する履歴情報の登録を行う。

【0035】

世代変更処理の後に、後述するステップ S 60 でプログラムアップデート処理を行い、さらにステップ S 70 で、図 3 (b) のように世代管理テーブル 5 の更新を行い、変更先世代区分 CHG を現世代区分 CUR へ変更し、新世代システム SYS 6 を構成する各ファイル世代 ID を登録して、以後の管理に供する。

なお、世代管理テーブル 5 の更新は、ステップ S 30、S 50 で行ってよい。

【0036】

図 6 は、上記したステップ S 30 の世代変更処理を示すフローチャートである。

ここでは、図3に示す現システム世代SYS4から変更先システム世代SYS5へ、世代変更する場合を例に説明する。

【0037】

ステップS301で、SYS4を構成するファイル世代IDとSYS5のファイル世代IDを、世代管理テーブル5を参照してファイルAから登録情報までファイル毎に比較する。
ファイルA、Fは両者のファイルID世代が一致しているので、以後の世代変更処理はパスされる。

【0038】

ファイル世代IDが一致しない他のファイルは、ステップS302で、変更先世代には当該ファイルがあるか判定する。

「無し」の場合、即ちSYS4にあってSYS5にないファイルEは、ステップS305で、ファイル登録管理手段3を動作させて記憶手段1のファイル群1aから削除する。

【0039】

一方、ステップS302で「有り」と判定されるファイルB、Dおよび登録情報は、ファイル登録管理手段3を指示してSYS5のファイル世代を履歴情報2a、2bから生成し、記憶手段1の現世代ファイル群1aのファイルDおよび登録情報1bと置換する。

同様に「有り」と判定されたファイルC(V1)は、現世代SYS4には使用していないが履歴情報2aには登録があるので、履歴情報2aから生成され、現世代ファイル群1aに追加される。

【0040】

SYS4とSYS5の全ファイルが終了するまで、ファイル毎に上記の処理を繰返し(ステップS306)、現世代システム情報1の更新が終了する。

なお、この後すぐに、世代管理テーブル5の更新を行うようにしてもよい。

【0041】

これによれば、現世代から変更先システム世代への世代変更が自動化されるとともに、ファイルの生成と置換の処理が最小化されて、世代変更におけるオーバヘッドを低減することができる。

【0042】

次に、上記したステップS50の新世代登録処理を、図7のフローチャートにしたがって説明する。
現世代システム情報1はステップS40により、図3(b)に示すSYS5からSYS6の各ファイルに更新されているものとする。

【0043】

まず、ステップS500で現世代システム情報1(SYS6)の各ファイルについて履歴情報の有無をチェックする。

履歴情報のある場合はステップS501で、その履歴情報2a、2bから最新世代ファイルを生成する。

つぎに、ステップS502で、この最新世代ファイルと現世代システム情報の該当ファイルを照合する。

照合は両ファイルの内部データについて行い、両ファイルが一致しているときは(ファイルB、C)、ステップS503で履歴情報の登録処理がパスされる。

【0044】

両ファイルのデータが一致していないとき(ファイルA、D、Gおよび登録情報)は、各々の最新ファイルIDを更新し(S504)、両ファイルの内部データの差分による履歴情報を、ファイル登録管理手段3により作成して、履歴情報記憶手段2に登録する(S505)。

また、世代管理テーブル5のSYS6の該当ファイル欄に、データに変更のないときは「→」、変更のあるときは新ファイルIDを設定し(S506)、その後、S501で生成した最新世代ファイルを削除する(S507)。

なお、ファイルGは新規の登録情報であり、履歴情報にはそのV1の全データが登録される。

【0045】

これによれば、履歴情報の登録が自動化されるので、多数のファイルを有するシステムの場合にも信頼性の高い版管理が可能になる。

また、更新のオーバヘッドも削減できる。

【0046】

このように、世代変更または新世代登録された現世代システム情報1のソースファイルは、その登録日付も同時に更新されるので、ステップS60のアップデート処理により、各段階の動的ファイルやプログラムが再登録される。

【0047】

ところで、上記した世代変更処理において、ソースファイルの変更はないがそのプログラム登録情報は変更される場合がある。

ステップS60は、このような場合にも自動的に実行されるアップデート処理を含むもので、図4に示す現世代SYS4から変更先世代SYS5へのプログラム登録情報を例に、図8のフローチャートにしたがって説明する。

なお、同図に示すステップS613の処理が、従来のプログラムアップデートに相当し、S600～S612はその前処理となる。

【0048】

まず、ステップS600で、変更先SYS5の登録情報IDに変更(変更および削除)があるか世代管理テーブル5でチェックする。

V1からV2に変更されているのでステップS601に進み、対象ソースプログラム(ここではファイルとプログラムは1対1に対応するものとする)が変更されているかテーブル5でチェックする。

ソースプログラム、即ちソースファイルが変更されている場合は、ステップS613に移行しての通常のアップデート処理が行われる。

【0049】

一方、図3のファイルAの場合は、SYS4とSYSの間に世代変更がないので、この場合は以下の処理が行われる。

ステップS602でプログラム登録情報管理テーブル5-1を参照し、当該ソースプログラムAに登録情報全部の削除があるか判定する。

図4(a)～(d)は、プログラムAにたいするプログラム登録情報の変更例を示し、同図(d)が全削除の例である。

この場合は、ステップS603～S605で、SYS4のファイルA(V4)の各段階のファイル/プログラムが削除され、それぞれの記憶手段に残存する不要情報が取り除かれる。

【0050】

つぎに、ステップS606でコンパイルオプションの変更があるかチェックし、図4(a)の例のようにオプテ

イマイズレベルが変更していれば、ステップS607でソースファイルAの登録日付を更新する。コンパイル情報の変更でなければ、ステップS608でリンクロードオプションが変更されているかチェックし、同図(b)のように変更があるときは(削除も含む)、ステップS609でオブジェクトファイル日付を更新する。つぎに、ステップS610で、実行プログラム登録オプションの変更をチェックし、同図(c)のように実行プライオリティの変更があれば、実行形式ファイルの登録日付を更新する。

【0051】

以上の処理をSYS5の全プログラムについて終了したときに(S612)、通常のプログラムアップデート(S613)が実行される。

同図(a)の場合はソースファイルAの日付が更新されているので、SYS4で登録したオブジェクトファイルAの日付と比較される。

この結果ソースファイル日付の方が新しいので、変更されたオプティマイズレベル2でコンパイルしたオブジェクトファイルAが再登録され、以下、実行形式ファイル、実行プログラムの再登録も連鎖的に行われる。

【0052】

同図(b)の場合は、オブジェクトファイル日付が更新されているので、SYS4で登録したオブジェクトファイルAを元に、標準のリンクロード仕様による実行形式ファイルの再登録と実行プログラムの再登録が実行される。

【0053】

同図(c)の場合は、実行形式ファイルの登録日付が更新されているので、オブジェクトファイルAと実行形式ファイルAはSYS4のままで、実行形式ファイルAからプライオリティ60による実行プログラムAの再登録が行われる。

【0054】

図9は、上記したS606～S611の変形例で、S607, S609, S611に対応するS607A, S609A, S611Aの各々は、日付の更新に代えて1段階下位のファイルまたはプログラムの日付を削除している。これによつても、S613のアップデート処理は図8と同じになる。

【0055】

図10は、プログラム登録情報に基づくプログラムアップデート処理を、実行手段を含んで概念的に説明した図で、コンパイラー61、リンクローダ62およびプロセス登録手段63(制御装置106の1機能)は、プログラムアップデート手段6の指示によって起動される。

【0056】

本実施例のプログラムアップデート処理によれば、通常のアップデートの実行に先立つて、プログラム登録情報に基づく所定の前処理を実行するので、プログラム登録情報のみの変更や削除に対しても、必要なアップデート処理が自動的に行なわれる。

これによつて、従来、ユーザが費していた多大な時間と手間が省力化され、変更システムの信頼性も向上できる。さらに、世代変更処理による業務の中止を大幅に短縮できるので、オンライン業務の速やかな再開による連続運転システムの操業率を向上できる。

【0057】

次に、分散システムのマスタ計算機から他計算機のシステムを変更する、本発明の第2の実施例について説明する。

【0058】

図11は、分散システムの構成と、マスタ計算機101aに接続される端末110の表示画面201を示す。表示中のシステム内世代情報202は、システムの各計算機(名称)にたいするシステム世代、計算機種別および管理システム世代を示している。

この例の計算機種別はTCSとHBSの2種類で、同一の種類は同一のソフトウェア構成を有し、TCSはノードA(101a)、HBSはノードE(101e)がマスタ計算機となって管理している。

【0059】

マスタ計算機101aは、第1の実施例で説明した自己の世代変更とともに、他の計算機のシステム変更に必要な情報を生成して、図1に示すシステム変更情報記憶手段11に収集し、この情報をシステム内の同一種別の他ノード101b～101dに配信して、各計算機のシステム変更を行うもので、図12はその概略フローを示している。

【0060】

ステップS701で、端末110からのシステム変更要求が受付られると、世代管理制御手段4は、ステップS702で現状のシステム内世代情報201を収集し表示する(S703)。この画面201には、変更計算機種別203と変更システム世代204の操作コマンド入力画面が表示されていて、ステップS704とS705でユーザの指示が受付られる。

以下では、種別がTCS、システム世代がSYS5の場合を例に、マスタ101aと他ノード101b～101dのシステム変更を例に説明する。

【0061】

ステップS706で、システム変更情報が収集され、記憶手段11に格納される。この収集が終了すると(マスタの変更終了)、ステップS707～709で、この変更情報をノードB、C、Dに順次、配送してシステム変更要求を行う。

この配送は、世代管理制御手段4の指示により、情報配布受手段10がシステム変更情報を所定のデータフォーマットに編集して後に行う。

【0062】

図13は、ステップS706に応じた、世代変更処理におけるシステム変更情報11の生成と収集の処理を示す。

これは、図6の世代変更処理のステップS304の生成ファイルの現世代情報1への置換と、S305の現世代ファイルの削除に対応して実行される。

すなわち、ステップS304A、Bで生成ファイルの名称と生成ファイルを、ステップS305Aで削除ファイルの名称をシステム変更情報記憶手段11に転送する。

これらの処理は、世代管理手段4の指示にしたがってファイル登録管理手段3が行う。

【0063】

図14は、プログラムアップデート処理におけるシステム変更情報11の収集処理を示す。すなわち、図8におけるステップS607、S609およびS611の日付更新処理に対応して、プログラムアップデートの開始点となる、コンパイルポイント、リンクロードポイントおよびプログラム登録ポイントをプログラムアップデート情報11bとして記憶する。

【0064】

図15は、システム変更情報のデータフォーマットで、(a)は、転送ファイル名称情報11a、(b)はプログラムアップデート情報11bについて示している。

【0065】

システム変更情報11は、情報配布送受手段10によって通信装置102を介して他計算機に配達される。マスターでない他計算機は、受信した変更情報を自己の情報配布送受手段10によってシステム変更情報記憶手段11に取り込む。

【0066】

さらに、マスターからのシステム変更要求によって、ファイル登録管理手段3が現世代システム情報記憶手段1を更新する。

すなわち、変更情報11の生成ファイル名称11aとそのソースファイルや登録情報11cを現世代システム情報の1a、1bに置換または追加し、削除ファイル名称11aに応じて現世代情報1aから不要なソースファイルを削除する。

さらに、プログラムアップデート情報1bに対応するファイル/プログラムについて、その指定のアップデート開始位置から、プログラムアップデート手段6による再登録を実行する。

【0067】

上記の例では、変更する計算機種別がTCSの1種であったが、複数の種別のシステム変更を連続的に実行することもできる。

また、システム内の世代管理を行うマスター計算機は、固定でなく他の計算機が代替することも可能である。

図16は、このような場合の処理フローを示す。

【0068】

各計算機は、自己の端末あるいはネットワークから変更計算機種別が受付されると、ステップS801で、その種別が自計算機と同一か判定する。

同一であれば、S802で自計算機がマスターか否か判定する。

マスターでなければ、S803でマスター計算機へのシステム変更要求を行う。

マスター計算機であれば、S804～S806でシステム変更情報の生成、収集を行い、他計算機に配布してシステム変更要求を行う。

これをシステム変更対象の全計算機種別が終了するまで繰返し行う(S807)。

【0069】

本実施例によれば、同一システム世代で稼働している複数計算機の変更情報の生成が1回ですみ、システム内の世代変更をオーバヘッド少なく高速に処理できる。

しかも、マスター計算機以外は現世代のプログラム群管理ですみ、履歴情報の管理が不要となる。

【0070】

なお、オンラインでのシステム変更が頻繁で、計算機間の転送情報量を低減したいときは、マスター以外の計算機に履歴情報を具備させ、変更情報11を差分情報で送るようにしてよい。

また、ネットワークによらず、システム変更に必要な情報を外部記憶装置を用いて他計算機にインストールすることも可能である。

【0071】

【発明の効果】

本発明のプログラム群世代管理方式によれば、変更先世代を指示するのみでシステムの世代変更が自動化でき、かつ、現世代と変更先世代のファイル世代が一致しないときのみファイルを生成するので、処理のオーバヘッドも削減できる。

【0072】

本発明のプログラムアップデート方法によれば、プログラム登録情報のみが変更する場合にも、プログラムアップデートの完全な自動処理ができるので、ユーザの手間と時間を大幅に省力化でき、プログラム群管理の信頼性を向上できる効果がある。

【0073】

本発明の分散処理システムによれば、多数の計算機の世代変更が1回の変更情報の生成で実行できるので、世代変更のための計算機の中止が短く、オンライン業務の処理効率を向上できる。

【図面の簡単な説明】

【図1】

本発明の一実施例である世代管理装置の構成を示す機能ブロック図である。

【図2】

本発明の一適用例である分散処理システムの構成図である。

【図3】

システム世代とファイル世代の相互関係を定義する世代管理テーブルである。

【図4】

システム世代のプログラム登録情報を定義するプログラム登録情報テーブルである。

【図5】

プログラム群世代管理の概略を示すゼネラルフローチャートである。

【図6】

世代変更処理の詳細を示すフローチャートである。

【図7】

新世代登録処理の詳細を示すフローチャートである。

【図8】

プログラムアップデート処理の詳細を示すフローチャートである。

【図 9】 プログラムアップデート処理の他の例を示すフローチャートである。

【図 10】 プログラムアップデート処理手段の構成を説明する説明図である。

【図 11】 本発明の第二の実施例である分散処理システムのシステム内世代情報を示す画面である。

【図 12】 他の計算機のシステム変更におけるマスタ計算機の処理を示すフローチャートである。

【図 13】 システム変更情報の転送ファイルの収集を示すフローチャートである。

【図 14】 システム変更情報のプログラムアップデート情報の収集を示すフローチャートである。

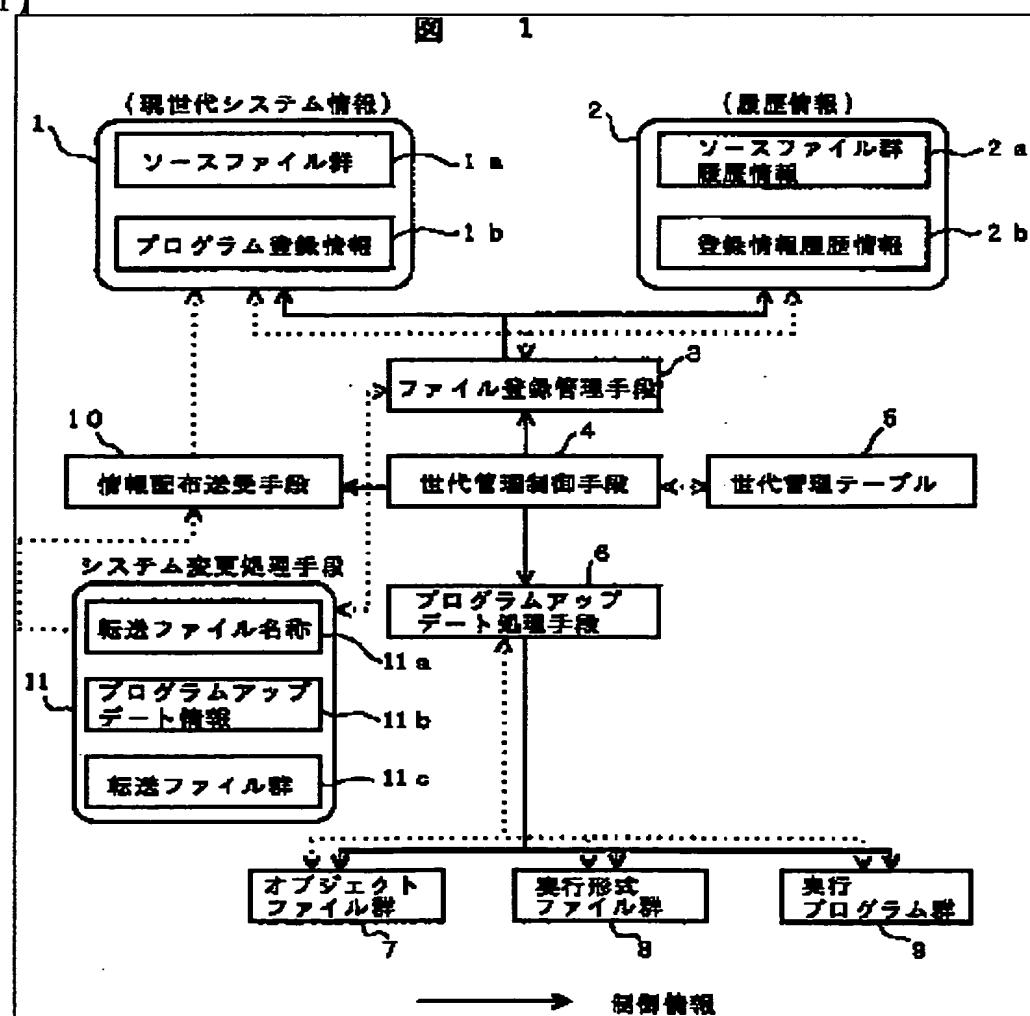
【図 15】 システム変更情報の構成を示すデータフォーマットである。

【図 16】 システム変更の他の処理例を示すフローチャートである。

【符合の説明】

1…現世代システム情報記憶手段、2…履歴情報記憶手段、3…ファイル登録管理手段、4…世代管理制御手段、5…世代管理テーブル、6…プログラムアップデート手段、7…オブジェクトファイル記憶手段、8…実行形式ファイル記憶手段、9…実行プログラム記憶手段、10…システム変更情報送受手段、11…システム変更情報記憶手段、101…計算機、102…通信装置、103…ネットワーク、104…世代管理装置、106…制御装置、107…入出力手段、109…磁気ディスク、110…端末装置、201…端末装置の表示画面。

【図 1】



【図15】

図 15

(a)転送ファイル名表

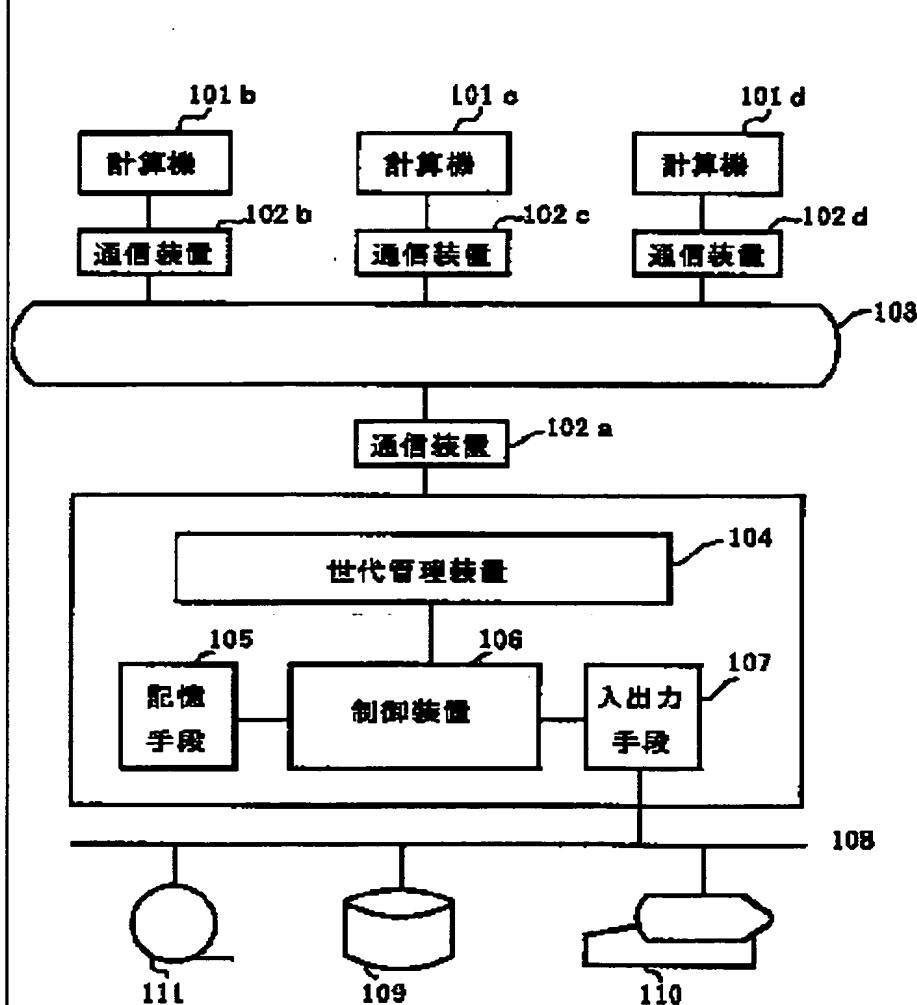
転送ファイル名	開封ファイル名
ファイルB、ファイルC	(なし)
ファイルD、登録情報	

(b)プログラムアップデート情報

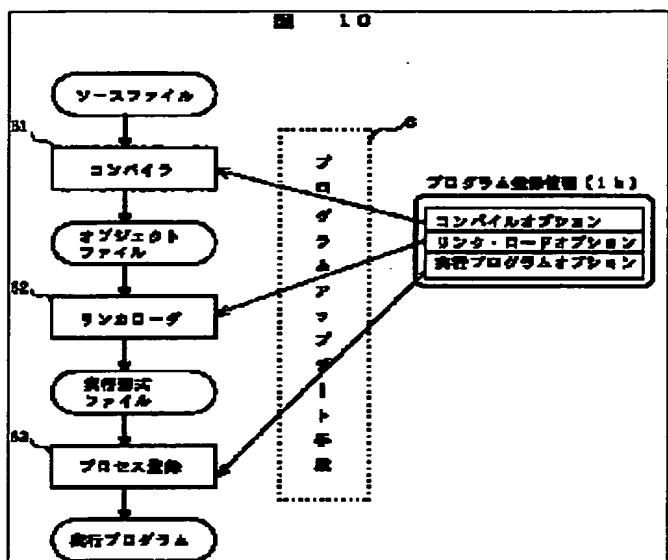
プログラム名	プログラムアップデートの開始ポイント		
	コンパイル	リントロード	プログラム更新
プログラムA	○		
プログラムB		○	
プログラムC			○

【図2】

図 2



【図10】



【図 3】

図 3

(a)

プログラム世代ID		SY81	SY82	SY83	SY84	SY85
区分		—	—	—	CUR	CHG
ファイル単位の世代ID	ファイルA	V1	V2	V3	V4	→
	ファイルB	V1	V2	V3	→	V4
	ファイルC	—	—	—	—	V1
	ファイルD	V1	→	→	V2	V3
	ファイルE	V1	→	→	→	—
	ファイルF	V1	→	→	→	→
	監査情報	V1	→	→	→	V2

(b)

プログラム世代ID		SY81	SY82	SY83	SY84	SY85	SY86
区分		—	—	—	—	—	CUR
ファイル単位の世代ID	ファイルA	V1	V2	V3	V4	→	V5
	ファイルB	V1	V2	V3	→	V4	→
	ファイルC	—	—	—	—	V1	→
	ファイルD	V1	→	→	V2	V3	V4
	ファイルE	V1	→	→	→	—	—
	ファイルF	V1	→	→	→	→	—
	ファイルG	—	—	—	—	—	V1
	監査情報	V1	→	→	→	V2	V3

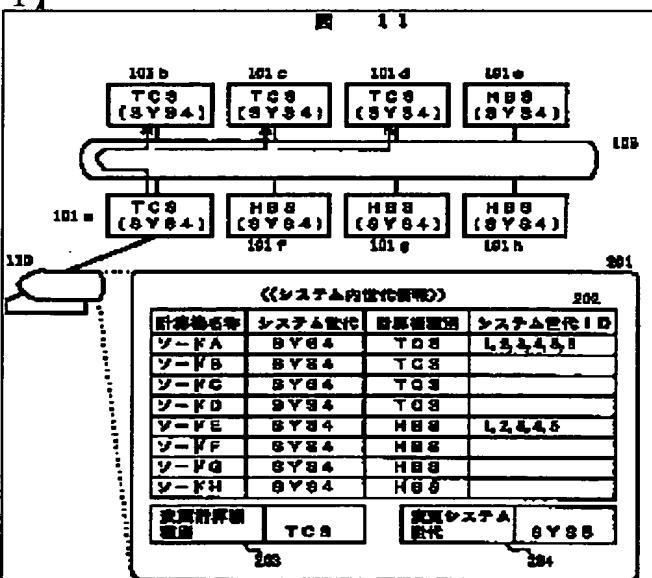
〈区分〉 CUR: 現世代
CHG: 直前世代

【図 4】

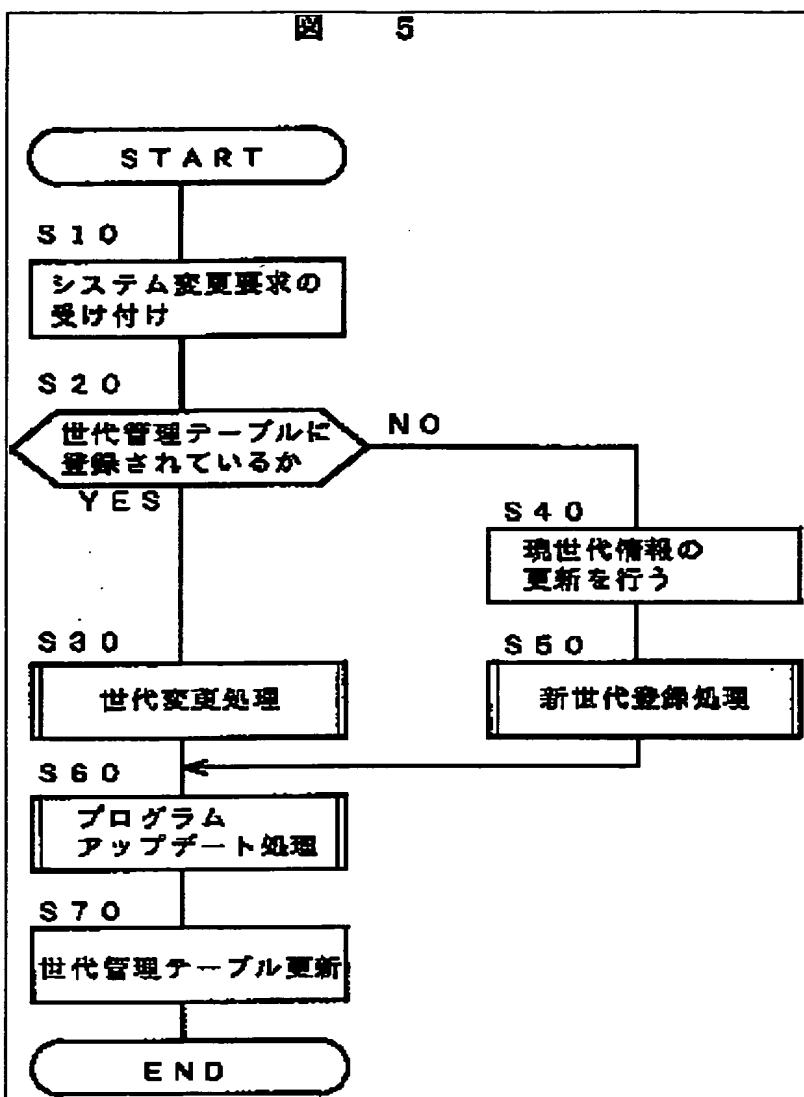
図 4

(a)	システム世代 ID	SY84	SY85	51
	コンパイルオプション (オプティマイズレベル)	3	2	
	リンク・ロードオプション	-C	-C	
	実行プログラムオプション (実行プライオリティ)	50	50	
(b)	システム世代 ID	SY84	SY85	51
	コンパイルオプション (オプティマイズレベル)	3	3	
	リンク・ロードオプション	-C	(無)	
	実行プログラムオプション (実行プライオリティ)	50	50	
(c)	システム世代 ID	SY84	SY85	51
	コンパイルオプション (オプティマイズレベル)	3	8	
	リンク・ロードオプション	-C	-C	
	実行プログラムオプション (実行プライオリティ)	50	60	
(d)	システム世代 ID	SY84	SY85	51
	コンパイルオプション (オプティマイズレベル)	3	(無)	
	リンク・ロードオプション	-C	(無)	
	実行プログラムオプション (実行プライオリティ)	50	(無)	

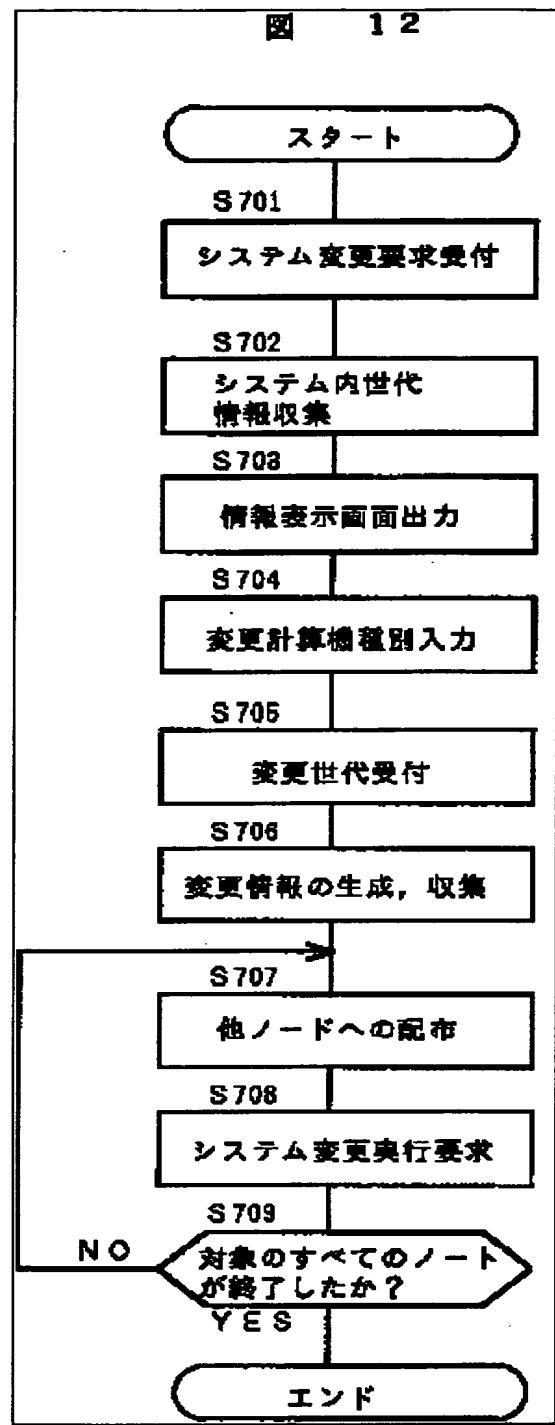
【図 11】



【図 5】

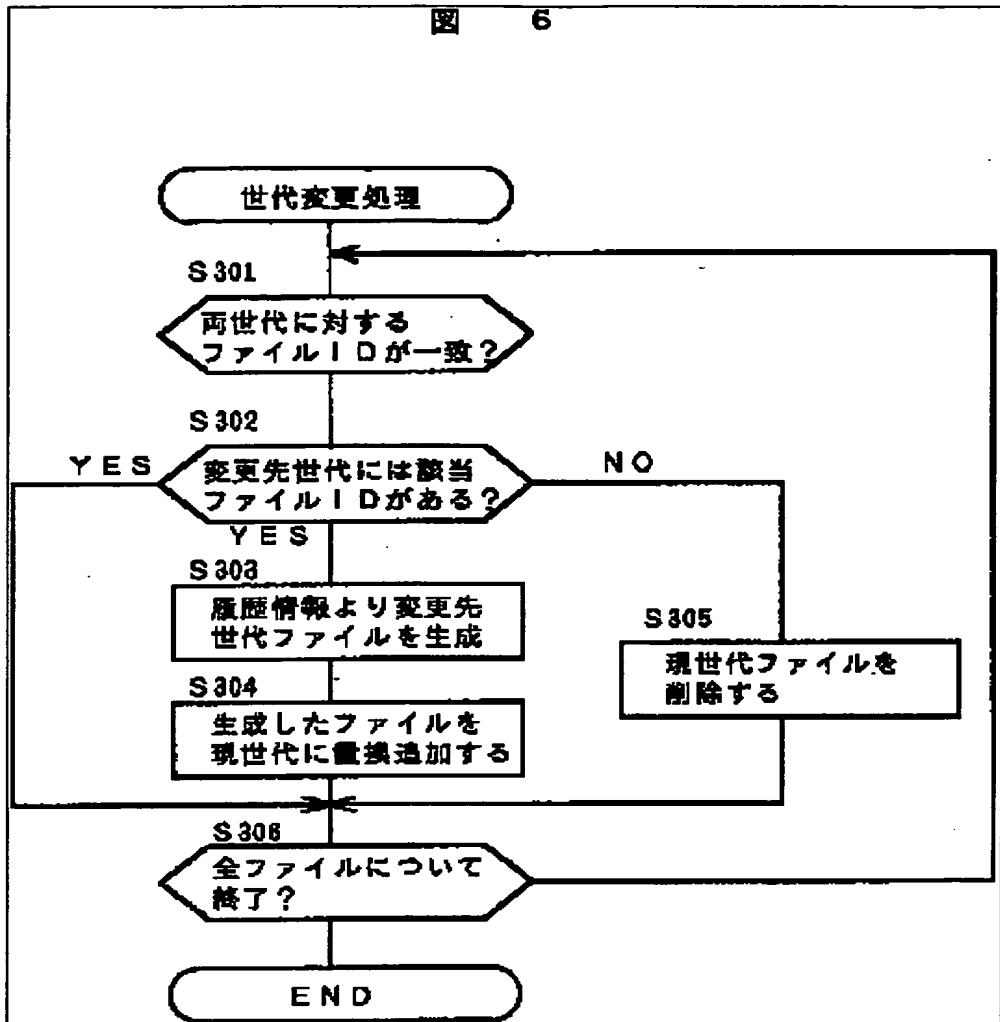


【図12】

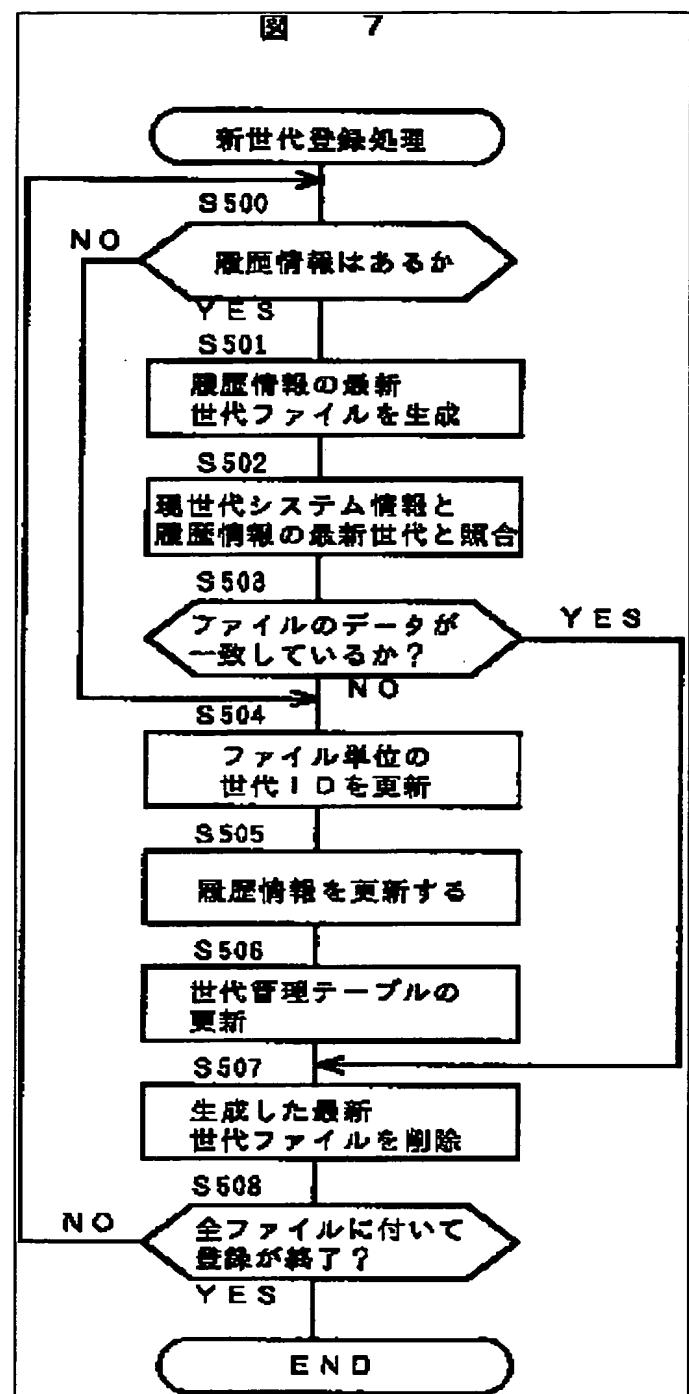


【図 6】

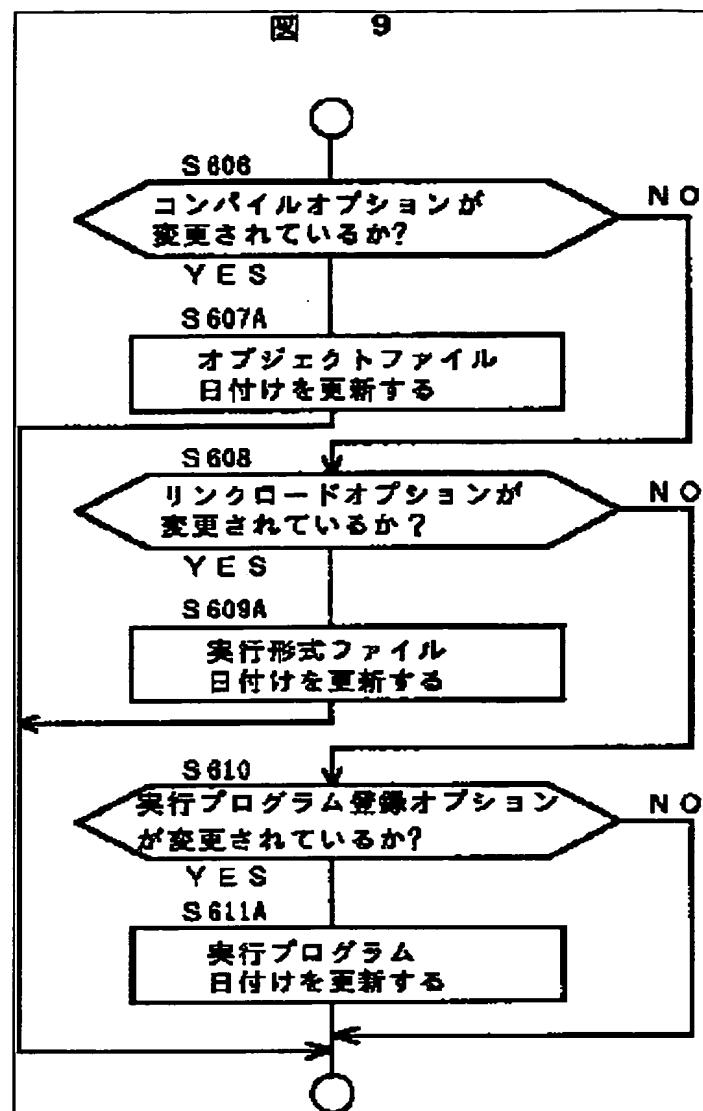
図 6

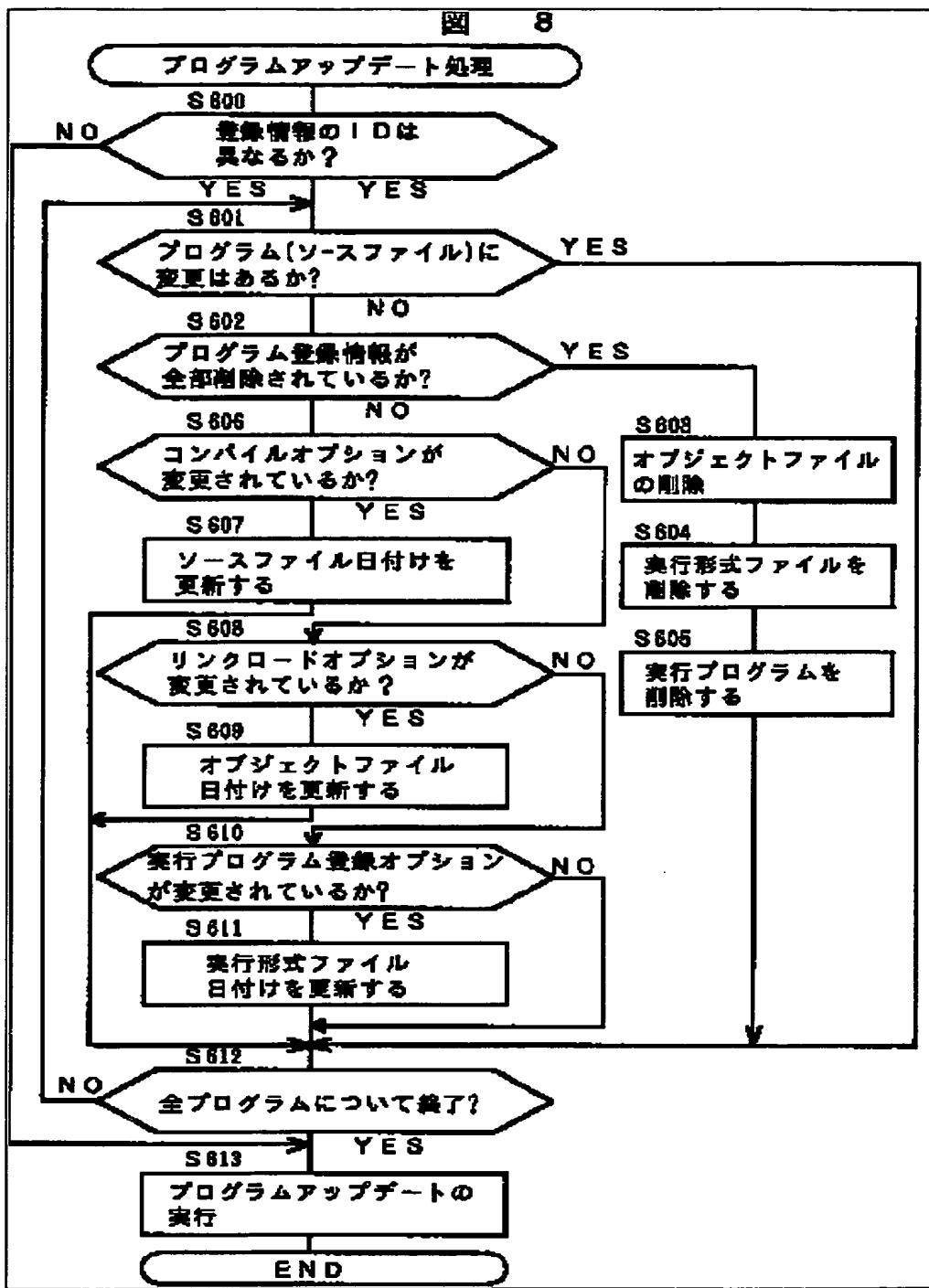


【図 7】



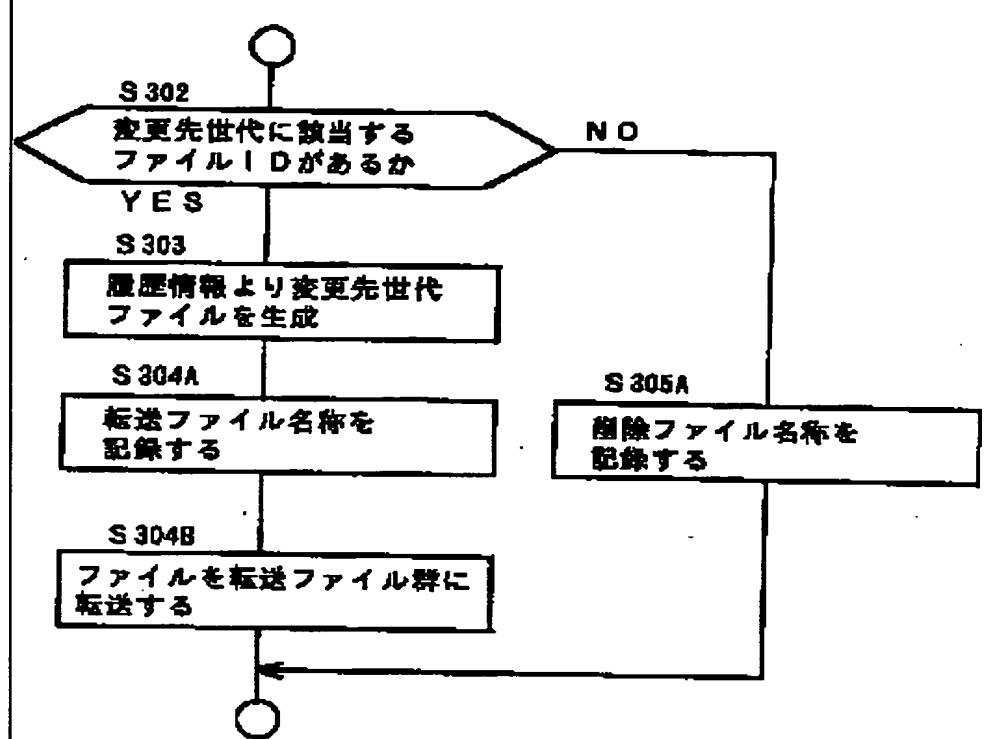
【図9】





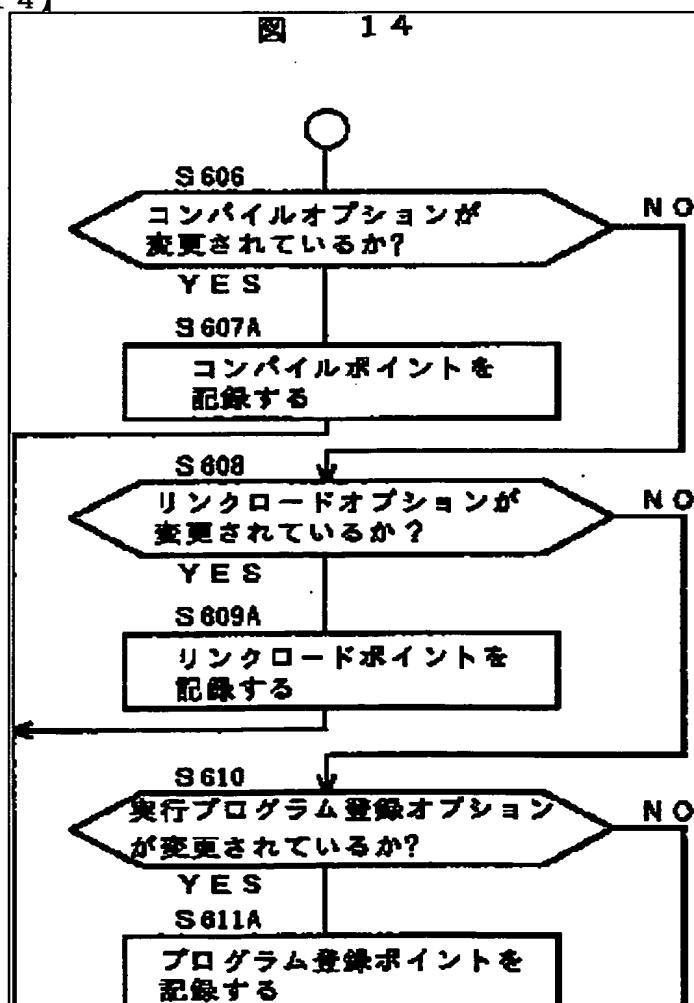
【図 13】

図 13



【図 14】

図 14



【図16】

図 16

